



Las Fuenteszuelas

Departamento de Informática

Informe:

Cursores de empleados.

Administración de Bases de Datos.

Ciclo Superior de Grado Superior en
Administración de Sistemas Informáticos.

Este documento ha sido realizado únicamente con fines educativos. Se ruega que el uso de los contenidos del mismo sean con el mismo fin, y que no sea copiado.

Jorge Navarrete Secaduras.

Jaén, 25 de Noviembre del 2021.



Contenidos:

1	Introducción	2
2	Manteniendo do while	2
3	Bucle while	3



1 Introducción

Este documento acompaña al .SQL donde se desarrolla el ejercicio 1 de la página 276 del libro, ya que me parecía demasiada información como para comentarla en el código.

Recomiendo leer este documento junto al código, ya que iré por orden explicándolo.

2 Manteniendo do while

La primera parte es arreglar el problema que nos genera el realizar este ejercicio con un bucle de tipo do while, ya que la última tupla se imprime por pantalla dos veces.

Para solucionarlo hay que controlar mediante un identificador de la tupla, que esta no se repita, por ello he decidido hacerlo con ROWID; aunque podría haber usado la propia clave primaria, he considerado más general y correcta esta opción.

Al necesitar dos columnas en la consulta, ya necesitamos cambiar la variable “v_apellido” del ejercicio a “v_emple” para volcar cada cupla en ella, por ello la declaro con rowtype. A continuación declaro la variable “rowid_anterior” de tipo varchar2 e inicializada con una letra cualquiera para eliminar basura de procedimientos anteriores.

Esta variable la usaré a continuación en el bucle, aunque antes hay que abrir el cursor y comenzar el bucle con el fetch del cursor into v_emple. Acto seguido se encuentra el IF clave de esta solución, donde indicamos que si el ROWID de la tupla anterior es diferente al de la tupla actual, imprima por pantalla la información.

En esa información he incluido que se imprima por pantalla el valor de nuestra variable “rowid_anterior” y el valor de ROWID actual para que se vea como funciona el cursor en pantalla.

En la primera tupla entrará en el IF, ya que la variable de “rowid_anterior” está inicializada con una letra aleatoria que no puede coincidir con una ROWID real. Después de este IF y de imprimir por pantalla la información, debemos hacer que la variable para guardar el ROWID anterior tenga el valor de la ROWID actual (es la actual pero realmente es la que ya hemos impreso por pantalla).

Esto hay que hacerlo usando la función “rowidtochar”, ya que ROWID es una pseudocolumna, y al no estar en el diccionario de datos como tal, no pude hacer que la variable de “rowid_anterior” fuese del tipo ROWID, y tampoco se puede guardar en una variable numérica o varchar directamente, hay que usar esa función.

Tras ello, volverá al inicio del bucle. En la última tupla, como el ROWID actual será el mismo que el anterior impreso por pantalla, no nos aparecerá, por lo que el resultado es este:

```
[2021-11-25 17:46:26] completed in 2 ms
[2021-11-25 17:46:26] 1.SANCHEZ a AAFFN2AABAAALJxAAA
[2021-11-25 17:46:26] 2.JIMENEZ AAFFN2AABAAALJxAAA AAFFN2AABAAALJxAAD
[2021-11-25 17:46:26] 3.GIL AAFFN2AABAAALJxAAD AAFFN2AABAAALJxAAH
[2021-11-25 17:46:26] 4.ALONSO AAFFN2AABAAALJxAAH AAFFN2AABAAALJxAAK
[2021-11-25 17:46:26] 5.FERNANDEZ AAFFN2AABAAALJxAAK AAFFN2AABAAALJxAAM
```

3 Bucle while

Aunque yo usaría un for, el bucle while funcionará mucho mejor en este caso, ya que cuando llegue a la última tupla, entrará al bucle con ella pero saldrá tras acabar esa última tupla, por lo que no hay que controlar su salida con tanto código.

Por ello el código es mucho más sencillo, volvemos a necesitar una única columna en la consulta, y la volcamos directamente en su variable correspondiente (v_apellido). Tra ello abrimos el cursor, hacemos el fetch y hacemos el bucle para que, mientras que encuentre tuplas, las imprima por pantalla.



Es importante hacer de nuevo el fetch dentro del while, ya que si no entrará en un bucle infinito con la primera tupla, con un for no tendríamos que controlar esto.

Así, con esa código más reducido, por pantalla nos dará el mismo resultado:

```
[2021-11-25 17:51:39] completed in 4 ms
[2021-11-25 17:51:39] 1.SANCHEZ
[2021-11-25 17:51:39] 2.JIMENEZ
[2021-11-25 17:51:39] 3.GIL
[2021-11-25 17:51:39] 4.ALONSO
[2021-11-25 17:51:39] 5.FERNANDEZ
```

NOTA: En mi IDE los procedimientos se ejecutan/llaman con "call", en otro IDE puede ser necesario usar "execute".

Jorge Navarrete, 2º ASIR.